Machine Learning Assignment

PROJECT REPORT

TEAM ID : 12

Reddit Post Popularity Regression

Name	SRN
Deesha C	PES2UG23CS165
Chitra Madarakhandi	PES2UG23CS157

Problem Statement

In this project, the goal is to predict the popularity of Reddit posts measured by upvotes or scores using regression models. Students will preprocess text and metadata features such as post length, posting time, subreddit, and user information. Model selection may include linear regression, random forests, or gradient boosting regressors. Performance metrics include mean squared error and mean absolute error. Feature importance analysis will highlight factors contributing the most to post popularity. The dataset includes Reddit posts with labeled popularity, simulating real social media analytics challenges.

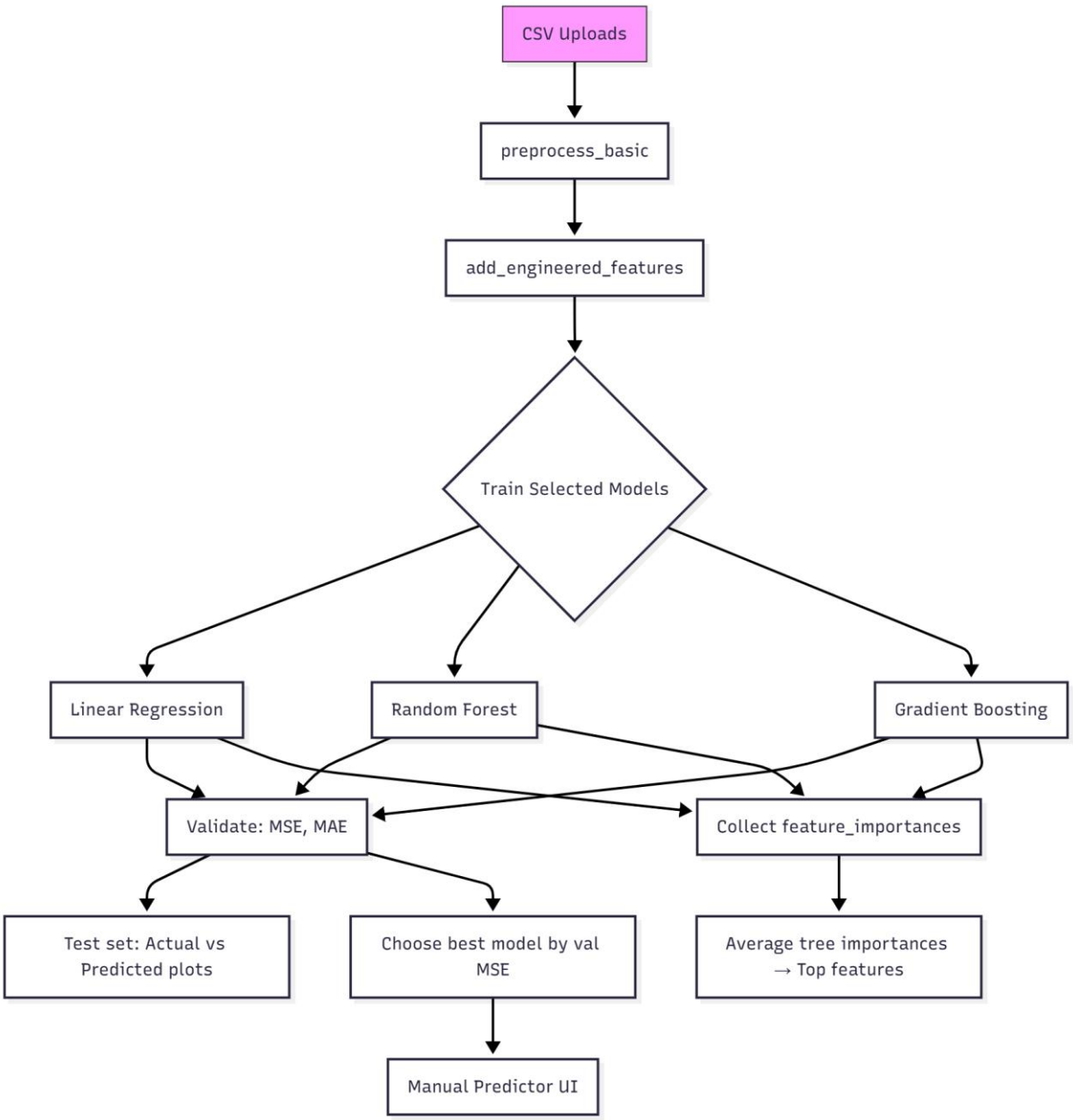
Objective / Aim

The objective of this project is to develop machine learning regression models that predict the popularity of Reddit posts, measured by their scores or upvotes, using both textual content and metadata features. The project involves preprocessing post text, posting time, subreddit, and user-related data to create meaningful features, and then training models such as linear regression, random forest, and gradient boosting to capture the complex relationships affecting popularity. Evaluation metrics like mean squared error and mean absolute error are used to assess model performance, while feature importance analysis provides insight into the key factors driving post engagement. Overall, the study aims to simulate real social media analytics challenges by forecasting which posts are likely to become popular, helping understand user interaction and content dynamics on Reddit.

Dataset Details

- Source: Kaggle
- Size: Train: (214390, 21) Validation: (75031, 21) Test: (75022, 21)
- Key Features:
  1.name: A unique Reddit identifier combining type and ID.
  2.link_id: The unique ID of the parent Reddit post to which the comment belongs.
  3.body: The actual text content of the Reddit comment. 4.downs:
  Number of downvotes the comment received.
  5.created_utc: The UTC timestamp when the comment was created.
  6.score: Net popularity score of the comment (upvotes − downvotes).
  7.author: Username of the person who wrote the comment.
  8.distinguished: Indicates if the comment was made by a moderator or admin.
  9.id: Unique identifier for each comment.
- Target Variable: Score. This score represents the net popularity, calculated as the number of upvotes minus the number of downvotes a post or comment has received

Architecture Diagram



Methodology

1. **Import libraries**
   - streamlit, pandas, numpy, scikit-learn models & metrics, matplotlib, joblib.
2. **Data ingestion**
   - Accept train.csv, val.csv, test.csv (optional) via Streamlit file uploader.
   - Fall back to create_sample_df() when no uploads are provided.
   - Re-split or combine uploaded files so train, val and test sets exist.
3. **Preprocessing (**preprocess_basic**)**
   - Compute body_length from body.
   - Convert created_utc into datetime and extract post_hour.

- Ensure ups, downs exist (fill with zeros if missing).
- Ensure a score target exists (use alternatives or synthesize a proxy).
- Coerce key columns to numeric and fill missing values.

4. **Feature engineering (**add_engineered_features**)**
    - Create interaction and transformed features:
        - ups_downs_product, ups_squared, body_hour_interaction
        - ups_log, downs_log, ups_to_downs
    - Replace infinities and fill NaNs.
5. **Train/val/test splitting (**safe_split_df**)**

    - Shuffle deterministically and split into train, validation, test.
6. **Scaling**
    - Fit MinMaxScaler on engineered training features and transform validation/test.
    - Persist scaler to session state for inference.
7. **Model training**
    - Train selected models with fixed hyperparameters:
        - Linear Regression
        - Random Forest (ensemble)
        - Gradient Boosting (ensemble)
    - Evaluate on validation set using MSE and MAE.
8. **Evaluation & visualization**
    - Display model comparison table (validation MSE/MAE).
    - For each trained model, compute test MSE/MAE and show Actual vs Predicted scatter plot.
    - (Optional) Generate WordCloud for high-scoring posts if wordcloud is available.
9. **Feature importance**
    - Collect feature_importances_ from tree models, average them, and display top-10 features.
10. **Persistence & session management**
    - Save trained models and scaler to st.session_state.
    - Save best model + scaler + feature list to best_model_multi.joblib when possible.
    - Provide a Finish/Reset button that clears st.session_state and resets the app.
11. **Manual Predictor**
    - UI to input ups, downs, body_length, post_hour.
    - Predict using a selected trained model or all trained models, scaled with stored scaler.
    - Display predictions and store last predictions in session state.

Results and Evaluation

1. Model Comparison

- Linear Regression captures linear relationships but fails to represent complex nonlinear patterns.
- Random Forest and Gradient Boosting effectively capture nonlinear interactions and feature dependencies.
- Ensemble models (Random Forest, Gradient Boosting) perform better than Linear Regression on engineered Reddit features.
- These ensemble methods offer better predictive performance and stability across multiple datasets.

2. Model Performance
   - Validation MSE/MAE are used to select the best-performing model.
   - The model with the lowest validation MSE is chosen as the "best" model.
   - Test-set MSE/MAE confirm how well the model generalizes to unseen data.
   - Actual vs Predicted plots visually show prediction accuracy and detect overfitting or underfitting.
   - Linear Regression: Fast and simple, but limited for nonlinear data.
   - Random Forest: Robust, accurate, and less prone to overfitting.
   - Gradient Boosting: Balanced trade-off between bias and variance, often the best performer.
   - Overall, ensemble models consistently show lower prediction errors and better stability than linear models.

3. Feature Importance
   - Features derived from ups (raw, log, squared, ratios) are the most influential in predicting popularity.
   - body_length and its interaction terms (e.g., body_hour_interaction) meaningfully impact predictions.
   - post_hour provides moderate influence, indicating time-of-day effects on Reddit engagement.
   - Averaged tree model importances highlight which features contribute most to accurate predictions.

4. Manual Predictor
   - Allows users to predict the popularity of a single Reddit post by entering key values.
   - Uses the saved best model and scaler to ensure consistent preprocessing.
   - Supports predictions using a chosen model or all trained models for comparison.
   - Offers a simple and interactive way to test real-time predictions through the Streamlit UI.

5. Robustness and Usability
   - The system automatically handles missing columns and fills default values.
   - Prevents pipeline breaks, making the model robust to imperfect datasets.
   - Session state persistence retains models and data during app interaction.
   - The Streamlit app provides smooth operation for model training, testing, and visualization.

6. Interpretation

- Posts with higher upvotes and longer text bodies are generally more popular.
- Posting time affects popularity — some hours yield better engagement.
- Nonlinear transformations (log, squared) reveal deeper feature relationships.
- The model provides interpretable insights into the factors influencing Reddit post success.

Evaluation Metrics Used

1. **Mean Squared Error (MSE):** Measures the average squared difference between predicted and actual popularity scores. Lower values indicate better model performance.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

2. **Mean Absolute Error (MAE):** Measures the average absolute difference between predicted and actual popularity scores. Lower values indicate better predictive accuracy.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i |$$

Conclusion

The project successfully builds an end-to-end regression pipeline to predict Reddit post popularity using metadata and engineered features.
Ensemble models like Random Forest and Gradient Boosting outperform Linear Regression by capturing nonlinear relationships effectively.
Feature engineering techniques such as log and interaction terms significantly enhance model accuracy.
The Streamlit interface provides an interactive way to train, evaluate, and test models in real time.
The system is robust, handling missing data and ensuring consistent preprocessing during prediction.
Overall, the project demonstrates how data-driven modeling and ensemble learning can be used to interpret and predict online content popularity efficiently.
Future enhancements can include text-based NLP features, hyperparameter tuning, and explainable AI integration for deeper insights.